

Embedded System for Real Color Composition in Experimental Physics

Sandro César Silveira Jucá¹, Renata Imaculada Soares Pereira², Gilmar de Pontes Barros Bezerra¹, Diego Lima Carvalho Gonçalves¹, Solonildo Almeida da Silva³

¹ Computing Department, Federal Institute of Ceará (IFCE), Brazil

Email: sandrojuca@ifce.edu.br, gilmardepontes@hotmail.com, zyhazz@gmail.com

² Electrical Engineering Department, Federal University of Ceará (UFC) - Pici Campus, Brazil

Email: renata@dee.ufc.br

³ Pedagogical Department, Federal Institute of Ceará (IFCE), Brazil

E-mail: solonildo@ifce.edu.br

Abstract—Software can be considered educational when properly contextualized in a relationship of teaching and learning. This paper describes an embedded system and an educational software developed and applied in experimental physics to real and virtual colors composition. The educational system developed allows student interaction by voice commands, which makes the assimilation process of red/green/blue (RGB) color formation concepts more didactic and playful. The application of educational software in experiments also allows to verify, in real time, the influence of programs change on the physical phenomena and stimulates the logical reasoning development and consequently the autonomy of the students, to the measure that can raise hypothesis, make modifications in the programming and take off conclusions from the practical results obtained.

Keywords—Android, Educational software, Experimental physics, RGB colors.

I. INTRODUCTION

On the interpretation of light phenomena interaction with matter and color perception by human eye, it is possible to consider three differences: the color perceived by the refraction of light on an object, like a yellow shirt that refracts a beam of light, with specific length; a color perceived as a determined electromagnetic wave emitted, as a yellow beam, with the wavelength of approximately 580 nm; and the color perceived by the interaction between eye and brain [1], that is, an object that emits light in the bands around green (wavelength around 520 nm) and around red (wavelength of 700 nm) can be interpreted as yellow by the sensation brain-eye. It's important to emphasize that the colors purple and gray, among others, are not spectral colors, that is, has no

specific wavelength, because they occur of multiple of others spectral colors as red, green and blue (RGB).

The human eye is capable of processing all of light's spectrum, continuous, since 400 to 700 nm, approximately. In that sense, Newton proved that white light when trespasses a crystal prism divides itself into multiple colored beams and determined that seven of those as the luminous visible spectrum, however only red, green and blue colors are the primary spectral colors that the eye-brain interaction physiologically captures. All the other colors are combinations formed by the brain from the average obtained by the proportion of these three colors [2].

Thus, for the human eye to perceive white light, the eye needs to receive all the visible light beams, as the sunlight, or simply receive the three primary spectral colors, RGB, in the same proportion. This principle is also utilized in other electronic device such as TV monitors, projectors, scanners and digital cameras.

On the experiment of the RGB formation, each color is defined by the red, green and blue proportions that composes it, as illustrated in Figure 1. For convenience, most digital systems today, as the educational software proposed, utilizes integers between 0 and 255 to specify those quantities. The number 0 indicates the lack of intensity and the number 255 indicates the maximum intensity of each color. Figure 1 illustrates the three primary colors of emission and the respective secondary colors yellow, magenta and cyan, obtained by the combination by pairs and equal to the proportion of primary colors and the color white obtained by the association of the three primary colors.

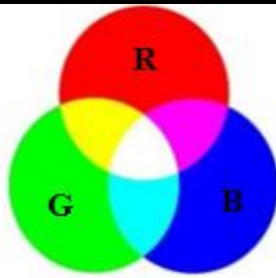


Fig 1: Process of primary colors formation by emission

On the other hand, it is currently common to develop low cost physics experiments [3] and using embedded systems [4] or microcontrollers [5] to facilitate the teaching-learning process. In this sense, the developed physics experiment aims to prove the composition of colors obtained in a virtual way. The emission of colors is done through the screen of an Android device, and at the same time, the colors are reproduced in a real way through voltage modulation applied to an RGB LED through the fundamental color ratios sent from the Android device to the microcontroller embedded system via Bluetooth. The proposed embedded system builds the same color emitted by the educational software screen on the Android device through an RGB LED.

II. MATERIALS AND METHODS

What designates a software as educational is the application on the teaching-learning process, in the sense that a software can be considered educational when adequately contextualized in a relationship of teaching-learning. The educational software is framed in educative and applicative [6].

2.1 Educational Software

Among the main characteristics of an educational software is the didactic character, that allows the construction of knowledge on certain area with or without the mediation of a professor. Alongside the significant increase in the number of students that seek professional education, there is also the increasing on the need for laboratories and didactic resources, as well as others physical resources to satisfy the learning of the students. Due to the incompatibility of the existing physical resources in relation to the demand, universities and educational institutions, didactic environments of simulation of real physical components are implemented [7]. That way, the professional learning is based mainly on the computational modelling of real systems, that posteriorly will be experienced in professional life. With that, the professional education is increasingly dependent on educational software.

An educational software can be conceptualized as a program that has resources projected with the

intention and purpose of being used in contexts of teaching-learning, being designed specifically [8]. These programs are applicable in different purposes beginning with concepts definition and going through skills development and ending with issues resolution.

On the category of educational software, comes those that are not developed with educational purposes, but are utilized for it. These are programs for general use utilized on teaching context and cognitive development, like for example, databases, compilers, text processors, electronic spreadsheets and graphical editors.

The educational software applications can be applied also for technological purpose [9]. In software applications with the technological purpose, the importance is given to the concepts related to practical application. The educational software proposed on this project can be utilized either with technological purpose or educative.

On this context, the experimental physics can be an efficient way for the application of interdisciplinarity and the Piagetian theory: the student is led to think on the essence of the problem, assimilating it, to posteriorly, accommodate it in his perspective of knowledge. The professor also stops being the only provider of knowledge to become the partner in the process of learning.

2.2 Developed Educational Tool SanUSB

On the intent of stimulating the construction of real automatic systems like the utilized on experimental physics, it was developed a free educational software called SanUSB to facilitate microcontrollers programming in educational projects and to stimulate the reflection and the development of student's autonomy on the elaboration of real projects.

SanUSB allows USB burning and execution of a C compiled program on a microcontroller, as also provides serial communication through the emulation of a serial COM port in a fast and efficient way from the moment that the microcontroller is connected directly to a computer via USB, as shown on Figure 2.

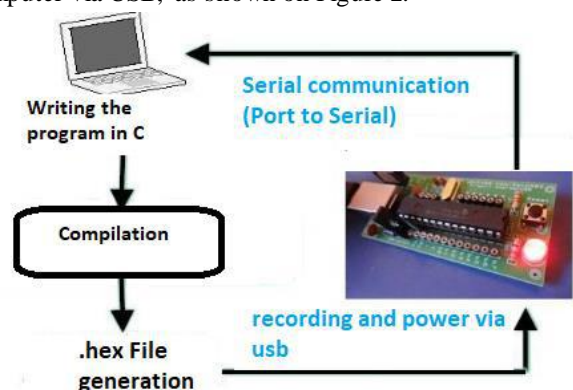


Fig 2: Programming process of the educational software SanUSB

A microcontroller is a low cost chip that contains internally peripherals of a dedicated computer such as a microprocessor, data memory, program memory, input and output pins, also as, USB interface in the case of more recent models, among others [10]. There are many advanced and creative ways of using them, keeping it low cost [11].

The educational software SanUSB is part of a tool composed by the software and basic circuits of a microcontroller PIC18Fxx5x, as illustrated on Figure 3.

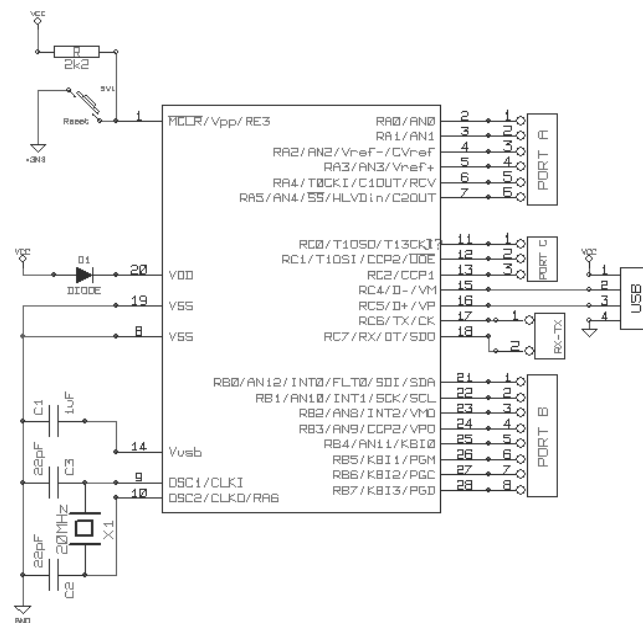


Fig 3: Basic microcontroller circuit of the educational tool SanUSB

SanUSB is registered by the Brazilian National Institute of Industrial Property (INPI) through the number 088503 as free software and is executable on Linux, Mac OSX and Windows™ operating systems, which are available for download and installing on the SanUSB group files [10].

This tool is composed of a pre-programmed bootloader on the microcontroller and an educational software application to burn the compiled codes on the memory of the microcontroller via USB.

The sending of the codes to the microcontrollers are normally made through specific commercial recording hardware. Through SanUSB, it is possible to burn and reburn the microcontroller directly from the USB port of any computer (even Raspberry Pi) accessing the free SanUSB Java interface as shown in Figure 4.

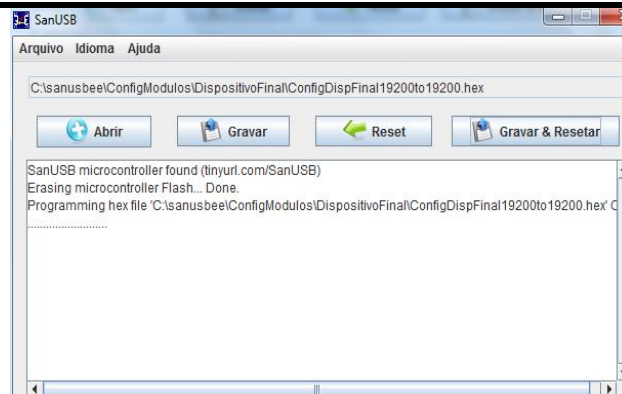


Fig 4: Free SanUSB Java interface for microcontroller burning

To program SanUSB circuit, it is only needed to press the button on pin 1, to connect the microcontroller to the computer through a USB cable, open the interface of the educational software application (Figure 4), select the “.hex” compiled file and burn the microcontroller with the code for the RGB color application experiment.

During project development, the power supply of the microcontroller comes from the USB port (5V). In the case of the microcontroller system be embedded, for example, in a robot or a system for data acquisition, it is necessary an external power supply, that can be a common battery of 9V or a cell phone charger.

Equipment projected and built for specific applications and for didactic purposes tend to be less expensive, having a better cost-benefit, for example, a Raspberry Pi, that even though it has more advanced capabilities like the interaction with more devices and data [12], it is more expensive and do not provide an environment as easy as the more specific one for assimilation and operation purposes.

The SanUSB tool is based on free software, that are developed from voluntary contributors through the Internet by a community that include developers across the world. This software offers performance enhancing, incentives creativity, allows dedicated applications and enables to locate and correct code mistakes faster than on proprietary software [13].

On that way, the utilization of free code programs represents a significant advantage for the continuous improvement of projects, without the dependency of a determined manufacturer or provider of the project. The free tool SanUSB also allows deuration directly through serial emulation (Communications Device Class - CDC protocol) via USB. This can be created in a quick and efficient way on the moment that the microcontroller is plugged directly to a PC, through a USB interface. That way, SanUSB is capable of suppressing:

- A specific equipment or circuit for burning a program

in the microcontroller, that is, it is not necessary either to take the microcontroller out of the embedded circuit to burn a new code or to use a conventional burner;

- TTL/RS-232 converter for serial bidirectional communication, because utilizes serial emulation via USB through the CDC protocol, that also allows the depuration of the program through printing via USB of the firmware variables;
- Power supply, as the power for the microcontroller comes from the USB port of the PC;
- External analog-to-digital converter (A/D), having in mind that the microcontroller has internally an A/D converter with up to 13 channels of up to 12 bits, depending on the chosen PIC18F model.
- Simulation software, considering that the practical application of the SanUSB coding and hardware mounting can be done in a fast and effective way on the development circuit board or on a matrix of contacts (breadboard).

In resume, among the features of the SanUSB free computational tool it is possible to emphasize the advantages:

- Programming in C language using different compilers;
- Free online support group to programming and projects development in Portuguese;
- Works on Linux, Mac OSX and Windows™ operating systems;
- The layout board allows the user full domain on the circuit, being able to easily realize maintenance, because no component is a surface mounted device (SMD);
- Reaches publics from beginners to high level designers;
- Didactic material provided allows self-learning [10].

The communication in experiments using free educational software can be realized not only via USB, but also with wireless communication protocols, like Bluetooth and WiFi. On the present proposed experiment, Bluetooth protocol was chosen. The name 'Bluetooth' has the origin from a Viking conqueror, called ^{VI...11}CATHODE (-) Bluetooth, that unified Denmark and Norway in the X century, symbolizing the union of different groups of people [14]. The idea of Bluetooth is to substitute the various proprietary solutions for communication with a pattern that can be adopted at a worldwide level. On Bluetooth transmission, the communication is made applying a spectral scattering technique with frequency leaps, denominated Frequency Hopping Spread Spectrum (FHSS).

Thus, the frequency is never the same, minimizing the interferences and making the existence of

another Bluetooth on the same geographical area possible [14]. The main requirements that guided the development of Bluetooth was the low consumption, low cost due to mass production of chips, the range of about 100 m (in the case of the model utilized in this project, the communication reaches up to 30 m) and voice and data transmission. Figure 5 illustrates the connection between SanUSB microcontroller development board and the Bluetooth module using four wires (Tx and Rx for serial communication; and Vcc and Gnd for power).

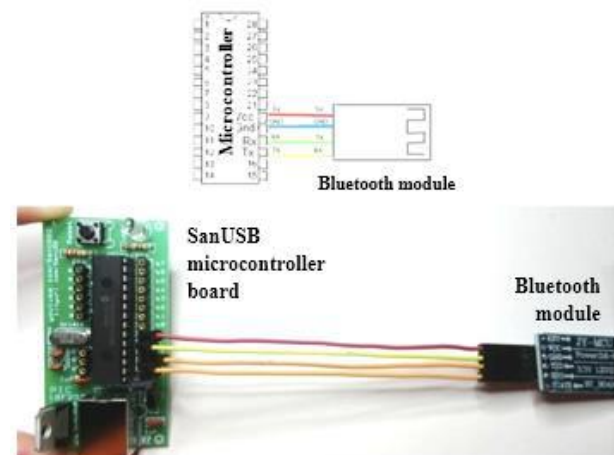


Fig 5: Illustration of the connection between SanUSB microcontroller board and Bluetooth module

On the other side, the RGB module used is built for this experiment, as can be seen on Figure 6 together with a RGB LED, showing the description of the connection pins. The RGB LED has R, G and B outputs that must be connected to digital pins like b0, b1 and b2 of the microcontroller, responsible for modulating in high frequency the width of the pulse (PWM) in each RGB LED pin. Thus, the SanUSB microcontroller board modulates the voltage applied in each of the RGB LED pins and, consequently, the intensity of the brightness for the composition of a desired real color. Figure 6 illustrates the RGB LED module developed for the proposed color composition experiment.



Fig 6: RGB LED module developed for the proposed experiment

For the proposed experiment, a C language (firmware) program was developed to receive information from Android devices via Bluetooth to manipulate the

RGB module pins, as it is shown on Figure 7. The free compiler utilized was the C18 in conjunction with the multi-platform environment MPLAB X. For achieving bigger diffusion of the embedded system proposed, it was also developed a program (sketch) for communication with the Arduino platform that can be accessed directly on the screen of the educational Android application (app), illustrated on Figure 8. Through the updating of the firmware it was also possible to alter the color composition through voice command. The experiment with voice recognition turned the contextualizing of the color formation theory more playful and interactive.

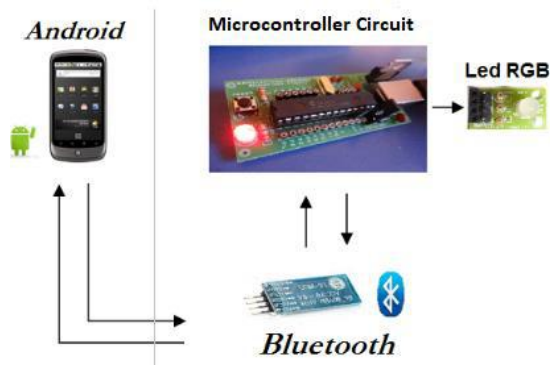


Fig 7: Illustration of the Bluetooth communication between the Android device and the SanUSB microcontroller circuit for RGB color composition

2.3 Educational Application for Android Devices

This subtopic describes how the free educational software denominated BT4SanUSB, developed and available freely for Android devices in the Google Play environment works. The application developed for Android allows to command the RGB color composition via Bluetooth utilizing the application's disc of colors or voice recognition. The voice recognition system utilized was developed by Google. This is based on Artificial Neural Networks (ANN), that is, a system of visual learning inspired by the human brain. On Figure 8 screens of the free educational Android app applied in experimental physics for real and virtual color composition are shown.

This application also does, by PWM control of the RGB LED via Bluetooth connected to the microcontroller, the real change of bright intensity, and permit that a determined color composition of an experiment be stored on the permanent memory of the application, pressing the button "Save". Beyond that, this application allows that the user change the name of the Bluetooth device directly from the Android device, inserting the name on the text box and pressing the button "Send name".

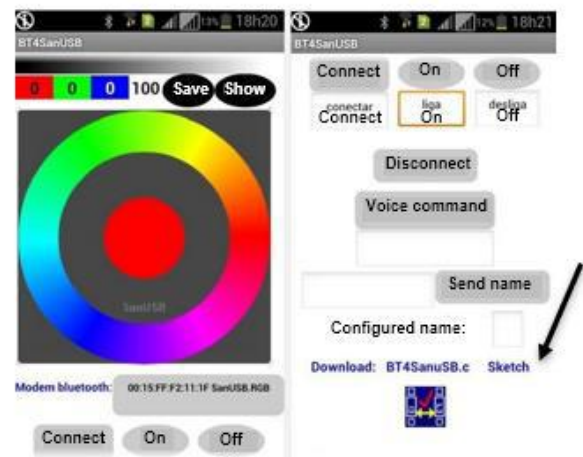


Fig 8: Screens of the educational Android application BT4SanUSB for Bluetooth control of RGB colors

The voice commands stored on the permanent memory of the Android device allows to correct failures on the recognizing software for a determined command. Although the answer found by the Android voice recognition software is different than the pronounced, that answer is still stored either way and when that voice command was pronounced again, there is a comparison with the stored word and in case of equal words, the command is executed.

2.4 Android Devices

One of the advantages of Android is that it is an operational system that works under Linux core, that is, a free software that allows developers to write software on the Java programming language to control the device via libraries developed by Google.

Beyond that, there is two main points: cost and portability. The Android devices are portable and low cost, which creates the increasing preference in relation to notebooks or conventional desktops. To endorse this affirmation, a research made by the CONECTA group and by the Worldwide Independent Network of Market Research (WIN), Brazil leads the use of smartphones and tablets in the world [15].

According to that research, the Brazilian average is 84 minutes a day using smartphones while the worldwide average is 74 minutes. In relation to tablets, the Brazilians use it for 79 minutes a day while the worldwide average registered by the research is 71 minutes. Even with these numbers and having the penetration of smartphones doubled since 2011 and 2012 going of 9% to 18%, Brazil continues far away from the worldwide average registered on the last year: 48% [15]. The research interviewed a total of a thousand people over 16 years old in Brazil, through the panel

CONECTAí. Beyond that, the research heard a total of 54 thousand in 54 countries in the world.

In relation to education environment, the use of notebooks is been switched to tablets either in public schools and private ones [16]. Considering this motivation, it was decided to use the free operating system Android in conjunction with the free software SanUSB.

2.4 Voice recognition

This application is also capable of triggering by voice commands the RGB LEDs or other devices connected to the microcontroller. The voice command can be recorded in the permanent memory of the application by holding the “Voice command” button for two seconds.

To control the BT4SanUSB through voice commands, it was utilized the library developed by Google already present on Android devices. The libraries of voice recognition can be on the Android device or in the cloud. From version 4.1, it is possible to make voice commands offline, that is, without having Internet connection.

Basically, all the applications that have voice recognition are split into two categories: limited vocabulary/many users or broad vocabulary/limited users. On the second, the program counts with a vocabulary of thousands of words and a degree of accuracy bigger or equal to 85% in recognizing. However, it has more accuracy, but for a limited number of users or even one person. This happens because it is necessary to train the application with a certain kind of voice or tone.

When Google developed the 4.1 version of Android mobile operating system, it implemented a series of changes in the way that the software recognizes human voice, based mainly on neural network, that is, a system of virtual learning based on how the human brain works.

In relation to the process of development and the basic functioning of the voice recognition algorithm, the error rate suffered 25% reduction in relation to previous versions of the Android operating system [17], making that more people talk in a more natural way with their smartphones, instead of dictating words as they were talking to robots. When a word is spoken to the Android voice recognition system, the spectrogram is sliced in small pieces and sent to eight computers located on Google servers. These pieces are processed accordingly with the neural networks models [17]. After that, the words are added to the software’s artificial intelligence like it had assimilated.

To realize the capturing of the words, are commonly adopted two types of frequency breaks: approximately eight thousand measures per second (digitalization on 8 kHz) or 44.100 times per second (44

kHz). The minor the break is, the bigger the accuracy of the sound captured. So, the Analog-Digital Converter (A/D) transforms these analog sound waves into digital, in a way that it can be analyzed by the device.

Then, the machine initiates the process of sound filtration, to separate noises from phonemes. On this step, multiple procedures happen, like separating the audio captured in different frequencies and do a normalization, that is, letting the same volume level. People also communicate in different speeds and so the computer needs to adjust the received sound to be in the same pronunciation time to be compared with the examples stored on the database.

With the filtered audio phonemes, the next step is for the computer to divide the sound into smaller pieces, with small fractions of time (hundredths or even thousandths of seconds). After, the program searches phonemes on the database libraries that are compatible with the local idiom and with what was recorded.

However, that task is more complex than it looks, because besides of existing approximately 34 phonemes on the Portuguese language (or 40 in English), the phonemes can vary according to regionalism and by the conjunction of a word. The same way, equal letters can have different pronunciations, according to the timing of it. The program needs then to realize an analysis of the phonemes and the context with the others around, following a complex model to compare them with words present in the library.

After recognizing each letter in the process explained earlier, the next step is to make them in order, to produce sentences. The main problem is that it can have a series of sequences that coincide with what was said by the person. The machine has then to analyze these sequences and verify those that make sense.

For this, the system does an analysis taking into consideration the probability of the known words to be positioned as they were understood. Some models also verify the dependency of each term of the sentence to confirm a correct meaning. In the app developed, after pressing the button “Voice command” and pronouncing a word, the word spoken is recognized and showed on the text box below the button.

III. RESULTS AND DISCUSSION

Figure 9 illustrates the operation of the embedded system during formation of real colors on the RGB LED and virtually on the Android device when the colors red, blue, green and all colors (white) are selected on the Android app BT4SanUSB.

The program for the microcontroller and the sketch for the Arduino can be downloaded directly from the screen of the BT4SanUSB application.

During the experiments of RGB colors formation, the educational software proved to be stable in relation to the data received to create the virtual and the real colors. The use of Bluetooth made possible that different users' profiles be able to install the application on Android devices and control it directly after pairing with the Bluetooth module connected to the SanUSB microcontroller board. Considering that the cost of a basic Android device is not so high (around US\$ 50.00 in Brazil), this project can be implemented in multiple educational institutions.



Fig 9: Operation of the BT4SanUSB app during formation of real colors on the RGB LED and virtually on the Android device

The visualization of the experiments through the comparison of real and virtual colors allowed the students to relate easily to what is seen and contextualize with the theoretic subject presented.

The voice command application also allowed to trigger loads through the predefined commands and after saying words like "On" and "Off" it is possible to make it work without pressing buttons on the application.

The SanUSB microcontroller circuits and Arduino are open source both in software and hardware, which facilitates the spreading of use of these tools through a big number of libraries and tutorials available on the Web.

IV. CONCLUSION

The educative software applications developed on this project favored the manipulation of real situations that students could make abstract conclusions through a reflection about the obtained results, contributing for the assimilation of the subjects and development of logical reasoning in a playful way.

Beyond that, the proposed educational software proved to be efficient in relation to programming and control by voice commands. In this sense, the voice

recognition of Android also proved to be efficient with the use of Bluetooth that made possible that different users could install the application on Android devices and control the RGB LED immediately after the pairing with the Bluetooth module.

The experiment of the embedded system for RGB color composition proposed was based on free software, with the goal of low-cost learning and interdisciplinarity of Physics and Computer Science. The use of the Bluetooth technology allowed the integration with the experiment of composition of colors in mobile devices, facilitating the interaction with students. In this way, the simplicity of construction of the experiments with embedded systems and use of free educational software contribute greatly for the creation of new didactic projects for experimental physics.

ACKNOWLEDGEMENT

The authors would like to thank the Federal Institute of Ceará (IFCE-Brazil), for the research scholarship and for making laboratory and equipment available.

REFERENCES

- [1] Nassau, K. (2001). The Physics and Chemistry of Color: The Fifteen Causes of Color. New York: Wiley & Sons, p. 454.
- [2] Pavoni, J. F., Neves-Junior, W. F. P., Spiropulos, M. A., de Araújo, D. B. (2014, October 3). An experimental setup for fluorescence measurement. Revista Brasileira de Ensino de Física [Online Exclusive]. Retrieved from: <http://www.scielo.br/pdf/rbef/v36n4/v36n4a18.pdf>
- [3] De Luca, R., Ganci, S. (2011, September 28). A measurement of g with a ring pendulum. Física com Arduino para iniciantes, Revista Brasileira de Ensino de Física. [Online Exclusive]. Retrieved from: <http://www.scielo.br/pdf/rbef/v33n3/01.pdf>
- [4] Cavalcante M. A., Tavoraro, C. R. C., Molisani, E. (2011, December 5). Physics with Arduino for beginners. Revista Brasileira de Ensino de Física [Online Exclusive]. Retrieved from: <http://www.sbfisica.org.br/rbef/pdf/334503.pdf>
- [5] Viana, J. C., Arnold, F. (2016, October 29). Development of an automated system for measuring electrical impedance. Revista Brasileira de Ensino de Física. [Online Exclusive]. Retrieved from: <http://www.scielo.br/pdf/rbef/v39n2/1806-1117-rbef-39-02-e2503.pdf>
- [6] Oliveira, C. C., Menezes, E. I., Moreira M. (2001). Ambientes Informatizados de Aprendizagem: produção e avaliação de software educativo. Campinas: Papirus, p. 144.
- [7] Gonçalves, R. N. S., de Andrade, J. E., Oliveira, R.

- A. P. (2017). A aprendizagem através de experimentos no Ensino de Física. *Revista Scientia Plena*. Doi:10.14808/sci.plena.2017.012709.
- [8] Sancho, J. M. (1998). Para uma Tecnologia educacional. Porto Alegre: ArtMed, p. 327.
- [9] Tajra, S. F. (2000). *Informática na Educação: novas ferramentas pedagógicas para o professor da atualidade*. São Paulo: Érica. p. 224.
- [10] SanUSB Group (2018, June 9). [Online Exclusive]. Retrieved from: <http://sanusb.org/>
- [11] Ravi C. Bathula & Sharath C. Inguva (2017). Autonomous mobile vehicle based on RFID technology using an ARM7 microcontroller. *International Journal of Advanced Engineering Research and Science* (ISSN : 2349-6495(P) | 246-1908(O)),4(5), 028-031. <https://dx.doi.org/10.22161/ijaers.4.5.6>
- [12] Ayub M. & Dr. K. Kalyani R. (2017). An IOT based Solar Integrated Home Security System by using GSM Module and Raspberry Pi. *International Journal of Advanced Engineering Research and Science* (ISSN: 2349-6495(P) | 2456-1908(O)),4(12), 195-199. <https://dx.doi.org/10.22161/ijaers.4.12.28>
- [13] Paulson, J.W., Succi, G., Eberlein, A. (2004). An Empirical Study of Open-Source and Closed-Source Software Products. *IEEE Transactions on Software Engineering*, 30(4), 246-256.
- [14] Muller, N. J. (2001). *Bluetooth Demystified*. New York: McGraw Hill, p. 396.
- [15] Portal G1. Brasil lidera o tempo de uso de smartphones e tablets (2004). [Online exclusive]. Retrieved from: <http://g1.globo.com/tecnologia/noticia/2013/04/uso-de-smartphone-e-tablet-e-maior-no-brasil-do-que-no-resto-do-mundo.html>.
- [16] Cronemberger, A. G., Soares, N. N. L., Gomes, M., Ferreira, F. C.L., Junior, J. G. S. L. (2017). Reflexões sobre o uso de tecnologias mobile como ferramenta auxiliar ao ensino. *Revista Scientia Plena*. Doi: 10.14808/sci.plena.2017.012720
- [17] Vanhoucke, V. How Google Retooled Android with help from your brain (2004). [Online exclusive]. Retrieved from: <http://www.wired.com/2013/02/android-neural>